

TECHNIQUE FOR INSTALLING A STATION DEVICE DRIVER

Cross-Reference to Related Applications

[0001] This application claims the benefit of U.S. provisional application Serial Number 60/454,539, filed March 13, 2003, entitled "Technique for Installing a Network Interface Card Driver," (Attorney Docket: 680-068us), which is herein incorporated by reference.

Field of the Invention

[0002] The present invention relates to telecommunications in general, and, more particularly, to wireless local area networks.

Background of the Invention

[0003] Figure 1 depicts a schematic diagram of network 100 in the prior art. Network 100 operates in accordance with a set of protocols, and comprises shared-communications medium 102 and telecommunication stations 101-1 through 101-K, wherein K is a positive integer. Network 100, for example, can be an Institute of Electrical and Electronics Engineers (IEEE) 802.11 local area network with shared-communications medium 102 representing the radio frequency spectrum with which stations 101-1 through 101-K communicate.

[0004] Host computing devices (*e.g.*, personal computers, personal digital assistants, *etc.*) use stations 101-1 through 101-K to communicate with each other or with other devices. Examples of other devices include printer servers, email servers, file servers, *etc.* Each of stations 101-1 through 101-K connects with a host computing device in the form of a peripheral card called a "network interface card" (NIC).

[0005] Wireless network providers are making wireless media, such as shared-communications medium 102, available in an increasing variety of places, such as coffee shops, fast food restaurants, business-class airport lounges, *etc.* Many of these wireless network providers require a user to have a pre-existing account or at least a properly-configured station. A potential user that wanders into an area served by a wireless network—for instance, to wait a few hours for a flight—might have handy a host computing

device, such as a laptop. The potential user, however, might not have handy a network interface card that enables convenient access to the particular network of interest.

Summary of the Invention

[0006] The present invention provides a host computing device user with convenient access to a network by enabling the efficient configuring of a telecommunications station with the host computing device. In the illustrative embodiment of the present invention, the telecommunications station stores device driver files and network-specific data in its memory (*e.g.*, flash memory, *etc.*). When the user connects the station to the host computing device, the host computing device installs a device driver from the device driver files received from the station. Once properly configured, the host computing device controls the station, through the device driver, to transmit data blocks into a shared-communications medium.

[0007] The network-specific data defines the network of interest by defining a network configuration and a security configuration. The host computing device receives a first portion of the network-specific data that enables the host computing device to properly configure itself to access the particular network of interest. In some embodiments, the host computing device displays the first portion of network-specific data to inform the user about the network. The network provider can control the amount of information about the network that the user sees by setting the content of the first portion of network-specific data ahead of time.

[0008] The station, in some embodiments, also stores a second portion of network-specific data that can neither be read nor overwritten by the host computing device. Only the station can use the second portion of data. The second portion can comprise restricted information such as encryption keys, authentication keys, *etc.*, allowing control over the host computing device's access to the network. Control might include setting the length of time for which the user is allowed to access the network.

[0009] In one operational scenario, the station, initially separated from the host computing device, is in the form of a network interface card. The user that enters the network purchases the network interface card on the spot (*e.g.*, from a vending machine, *etc.*). To facilitate the user's experience, installation takes advantage of plug-and-play in combination with an AutoRun file stored in the station. The station is self-installing, in that no other media is required to enable the station to function on the host computing device,

even where the host computing device has had no station device driver previously installed. The user does not necessarily have to enter network-specific parameters, since the network configuration and security configuration are pre-stored in the network interface card.

[0010] An illustrative embodiment of the present invention comprises: transferring a device driver file and a first portion of network-specific data from a station to a host computing device; installing at the host computing device a device driver that is represented by the device driver file; and transmitting a data block into a shared-communications medium that constitutes a network, wherein the host computing device generates the data block and wherein the host computing device uses the device driver to transfer the data block to the station; wherein the first portion of network-specific data defines the network.

Brief Description of the Drawings

[0011] Figure 1 depicts a schematic diagram of network 100 in the prior art.

[0012] Figure 2 depicts a schematic diagram of network 200 in accordance with the illustrative embodiment of the present invention.

[0013] Figure 3 depicts a block diagram of the salient components of station 201-*i* in accordance with the illustrative embodiment of the present invention.

[0014] Figure 4 depicts a block diagram of the salient components of host computing device 203-*i* in accordance with the illustrative embodiment of the present invention.

[0015] Figure 5 depicts a flowchart of the illustrative embodiment of the present invention.

Detailed Description

[0016] Figure 2 depicts a schematic diagram of network 200 in accordance with the illustrative embodiment of the present invention. Network 200 operates in accordance with a set of protocols (e.g., IEEE 802.11, etc.) and comprises telecommunications stations (or just "stations") 201-1 through 201-*L*, wherein *L* is a positive integer, and shared-communications medium 202, interconnected as shown. For example, network 200 can be a wireless local area network. It will be clear to those skilled in the art how to make and use shared-communications medium 202.

[0017] Station 201-*i*, for *i*=1 to *L*, is capable of receiving data blocks from host computing device 203-*i* and transmitting over shared-communications medium 202 data

frames comprising the data blocks received from host computing device 203-*i*.

Station 201-*i* is also capable of receiving data frames comprising data blocks from shared resource 202 and sending to host computing device 203-*i* the data blocks. It will be clear to those skilled in the art, after reading this specification, how to make and use station 201-*i*.

[0018] Figure 3 depicts a block diagram of the salient components of station 201-*i* in accordance with the illustrative embodiment of the present invention. Station 201-*i* comprises receiver 301-*i*, processor 302-*i*, memory 303-*i*, and transmitter 304-*i*, interconnected as shown.

[0019] Receiver 301-*i* is a circuit that is capable of receiving frames from shared resource 202, in well-known fashion, and of forwarding them to processor 302-*i*. It will be clear to those skilled in the art how to make and use receiver 301-*i*.

[0020] Processor 302-*i* is a general-purpose processor that is capable of performing the tasks described below and with respect to Figure 5. It will be clear to those skilled in the art, after reading this specification, how to make and use processor 302-*i*.

[0021] Memory 303-*i* is capable of storing programs and data used by processor 302-*i*. Memory 303-*i* also stores one or more device driver files. The device driver is the software that host computing device 203-*i* uses to interact with station 201-*i*. The device driver and associated files are to be described later. At least a portion of memory 303-*i*, in some embodiments, comprises flash memory, read-only memory, programmable read-only memory, magnetic disk memory, or compact disk memory.

[0022] Memory 303-*i* also stores network-specific data. The set of network specific data describes network 200. The network-specific data, in some embodiments, define a network configuration and a security configuration. The network configuration addresses the aspects of network identification and access control. In some embodiments, the network identification comprises an IEEE 802.11 basic service set identifier (BSSID), as is known in the art. The security configuration addresses the aspects of authentication and privacy (e.g., through encryption, etc.).

[0023] Memory 303-*i*, in some embodiments, also stores an AutoRun file, a Setup executable file, or both. These files are to be described later.

[0024] It will be clear to those skilled in the art, after reading this specification, how to make and use memory 303-*i*.

[0025] Transmitter 304-*i* is a circuit that is capable of receiving frames from processor 302-*i*, in well-known fashion, and of transmitting them on shared resource 202. It will be clear to those skilled in the art how to make and use transmitter 304-*i*.

[0026] Figure 4 depicts a block diagram of the salient components of host computing device 203-*i* in accordance with the illustrative embodiment of the present invention. Host computing device 203-*i* comprises processor 401-*i*, memory 402-*i*, and display 403-*i*, interconnected as shown.

[0027] Host processor 401-*i* is a general-purpose processor that is capable of performing the tasks described below and with respect to Figure 5. It will be clear to those skilled in the art, after reading this specification, how to make and use host processor 401-*i*.

[0028] Host memory 402-*i* is capable of storing programs and data used by host processor 401-*i*. It will be clear to those skilled in the art how to make and use host memory 402-*i*.

[0029] Display 403-*i* is capable of displaying information received from station 201-*i*. Display 403-*i*, in embodiments, displays a portion of the network-specific data from station 201-*i*. In some embodiments, display 403-*i* is not present, such as where host computing device 203-*i* is feature-limited or when the user of host computing device 203-*i* does not need to have access to the network-specific data. It will be clear to those skilled in the art, after reading this specification, how to make and use display 403-*i*.

[0030] Figure 5 depicts a flowchart of the salient tasks performed by the illustrative embodiment of the present invention. It will be clear to those skilled in the art which tasks depicted in Figure 5 can be performed simultaneously or in a different order than that depicted.

[0031] At task 501, host computing device 203-*i* requests one or more device driver files from station 201-*i* in well-known fashion. This is triggered, for example, when host computing device 203-*i* senses that it has become electrically coupled with station 201-*i* after a user has plugged in a network interface card into a card slot of a laptop. It will be clear to those skilled in the art how to sense the presence of station 201-*i* when plugged into host computing device 203-*i*.

[0032] In some embodiments, the request for one or more device driver files is generated in well-known fashion from an executable file (e.g., setup.exe, etc.) stored in memory 303-*i* of station 201-*i*. Host computing device 203-*i* runs the executable file—in

some embodiments, after reading an AutoRun file in well-known fashion, also stored in memory 303-*i*.

[0033] At task 502, station 201-*i* transfers the device driver files to host computing device 203-*i*. Station 201-*i* also transfers a first portion of network-specific data. The first portion of network-specific data can comprise parameters provided to host computing device 203-*i* for the purpose of providing status, general configuration information, diagnostic information, administration information, etc. It will be clear to those skilled in the art how to select which parameters constitute the first portion of network-specific data.

[0034] In some embodiments, station 201-*i* refrains from transferring a second portion of network-specific data. The second portion can comprise at least one encryption key, at least one authentication key, or both, and is unreadable by host computing device 203-*i*. The second portion is unreadable by host computing device 203-*i* to control, for example, how host computing device 203-*i* uses network 200. It will be clear to those skilled in the art how to make and use station 201-*i* so that the second portion of network-specific data cannot be read or overwritten by host computing device 203-*i*.

[0035] At task 503, in some embodiments, host computing device 203-*i* displays on display 403-*i* the first portion of network-specific data. For example, display 403-*i* might show basic status such as the name of the present network and whether or not the signal strength is sufficient. At the other extreme, display 403-*i* might show most or all of the network configuration, the security configuration, or both. It will be clear to those skilled in the art which network-specific data to display.

[0036] In some embodiments, station 201-*i* continues to transfer network-specific data to host computing device 203-*i* in well-known fashion. The data transferred can comprise parameters that change over time, such as the signal strength and the data rate over shared-communications medium 202. It will be clear to those skilled in the art how to select network-specific data that station 201-*i* transfers to host computing device 203-*i* on an ongoing basis.

[0037] At task 504, host computing device 203-*i* installs the device driver that allows host computing device 203-*i* to use station 201-*i*. The device driver files represent the device driver. Note that the device driver software installed into host computing device 203-*i* can be in a different form than that of the device driver files originally stored at station 201-*i*. In some embodiments, installing the device driver requires that the user specify selections for particular options (e.g., country selection, network name, channel

selection, etc.). In other embodiments, installing does not require any user intervention. For example, when a user acquires a network interface card from a vending machine for use in the immediate area (e.g., at an airport, etc.), the user might not either know or care about the network specifics, other than what is advertised on the vending machine by the network provider.

[0038] At task 505, host computing device 203-*i* is ready to use station 201-*i*. Host computing device 203-*i* generates a data block in well-known fashion. An application program (e.g., an email reader/sender, etc.) that needs to send a message to another device via network 200 can generate the data block.

[0039] At task 506, host computing device 203-*i* transfers the data block to station 201-*i* in well-known fashion.

[0040] At task 507, station 201-*i* transmits the data block into shared-communications medium 202 in well-known fashion using the network configuration defined by the network-specific data. Station 201-*i* can encrypt the data block in well-known fashion using the security configuration (e.g., encryption keys, etc.) also defined by the network-specific data.

[0041] It is to be understood that the above-described embodiments are merely illustrative of the present invention and that many variations of the above-described embodiments can be devised by those skilled in the art without departing from the scope of the invention. It is therefore intended that such variations be included within the scope of the following claims and their equivalents.

[0042] What is claimed is: